Homerwork 2

ORI STERN

5/21/23

# Data Visualisation - Exploration

Now that you’ve demonstrated your software is setup, and you have the basics of data manipulation, the goal of this assignment is to practice transforming, visualising, and exploring data.

# Mass shootings in the US

In July 2012, in the aftermath of a mass shooting in a movie theater in Aurora, Colorado, [Mother Jones](https://www.motherjones.com/politics/2012/07/mass-shootings-map/) published a report on mass shootings in the United States since 1982. Importantly, they provided the underlying data set as [an open-source database](https://www.motherjones.com/politics/2012/12/mass-shootings-mother-jones-full-data/) for anyone interested in studying and understanding this criminal behavior.

## Obtain the data

Rows: 125  
Columns: 14  
$ case <chr> "Oxford High School shooting", "San Jose VTA shoo…  
$ year <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2020, 2020, 2…  
$ month <chr> "Nov", "May", "Apr", "Mar", "Mar", "Mar", "Mar", …  
$ day <dbl> 30, 26, 15, 31, 22, 16, 16, 26, 10, 6, 31, 4, 3, …  
$ location <chr> "Oxford, Michigan", "San Jose, California", "Indi…  
$ summary <chr> "Ethan Crumbley, a 15-year-old student at Oxford …  
$ fatalities <dbl> 4, 9, 8, 4, 10, 8, 4, 5, 4, 3, 7, 9, 22, 3, 12, 5…  
$ injured <dbl> 7, 0, 7, 1, 0, 1, 0, 0, 3, 8, 25, 27, 26, 12, 4, …  
$ total\_victims <dbl> 11, 9, 15, 5, 10, 9, 4, 5, 7, 11, 32, 36, 48, 15,…  
$ location\_type <chr> "School", "Workplace", "Workplace", "Workplace", …  
$ male <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, T…  
$ age\_of\_shooter <dbl> 15, 57, 19, NA, 21, 21, 31, 51, NA, NA, 36, 24, 2…  
$ race <chr> NA, NA, "White", NA, NA, "White", NA, "Black", "B…  
$ prior\_mental\_illness <chr> NA, "Yes", "Yes", NA, "Yes", NA, NA, NA, NA, NA, …

| column(variable) | description |
| --- | --- |
| case | short name of incident |
| year, month, day | year, month, day in which the shooting occurred |
| location | city and state where the shooting occcurred |
| summary | brief description of the incident |
| fatalities | Number of fatalities in the incident, excluding the shooter |
| injured | Number of injured, non-fatal victims in the incident, excluding the shooter |
| total\_victims | number of total victims in the incident, excluding the shooter |
| location\_type | generic location in which the shooting took place |
| male | logical value, indicating whether the shooter was male |
| age\_of\_shooter | age of the shooter when the incident occured |
| race | race of the shooter |
| prior\_mental\_illness | did the shooter show evidence of mental illness prior to the incident? |

## Explore the data

### Specific questions

* Generate a data frame that summarizes the number of mass shootings per year.

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"))

Rows: 125 Columns: 14  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (7): case, month, location, summary, location\_type, race, prior\_mental\_i...  
dbl (6): year, day, fatalities, injured, total\_victims, age\_of\_shooter  
lgl (1): male  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Take a look at the data  
glimpse(mass\_shootings)

Rows: 125  
Columns: 14  
$ case <chr> "Oxford High School shooting", "San Jose VTA shoo…  
$ year <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2020, 2020, 2…  
$ month <chr> "Nov", "May", "Apr", "Mar", "Mar", "Mar", "Mar", …  
$ day <dbl> 30, 26, 15, 31, 22, 16, 16, 26, 10, 6, 31, 4, 3, …  
$ location <chr> "Oxford, Michigan", "San Jose, California", "Indi…  
$ summary <chr> "Ethan Crumbley, a 15-year-old student at Oxford …  
$ fatalities <dbl> 4, 9, 8, 4, 10, 8, 4, 5, 4, 3, 7, 9, 22, 3, 12, 5…  
$ injured <dbl> 7, 0, 7, 1, 0, 1, 0, 0, 3, 8, 25, 27, 26, 12, 4, …  
$ total\_victims <dbl> 11, 9, 15, 5, 10, 9, 4, 5, 7, 11, 32, 36, 48, 15,…  
$ location\_type <chr> "School", "Workplace", "Workplace", "Workplace", …  
$ male <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, T…  
$ age\_of\_shooter <dbl> 15, 57, 19, NA, 21, 21, 31, 51, NA, NA, 36, 24, 2…  
$ race <chr> NA, NA, "White", NA, NA, "White", NA, "Black", "B…  
$ prior\_mental\_illness <chr> NA, "Yes", "Yes", NA, "Yes", NA, NA, NA, NA, NA, …

# Using dplyr to group the data by year and count the number of cases  
mass\_shootings\_per\_year\_df <- mass\_shootings %>%  
 group\_by(year) %>%  
 summarise(Number\_of\_Mass\_Shootings = n())  
  
# Print the resulting data frame  
print(mass\_shootings\_per\_year\_df)

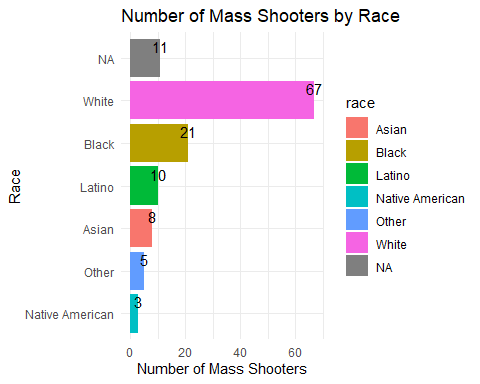
# A tibble: 37 × 2  
 year Number\_of\_Mass\_Shootings  
 <dbl> <int>  
 1 1982 1  
 2 1984 2  
 3 1986 1  
 4 1987 1  
 5 1988 1  
 6 1989 2  
 7 1990 1  
 8 1991 3  
 9 1992 2  
10 1993 4  
# ℹ 27 more rows

* Generate a bar chart that identifies the number of mass shooters associated with each race category. The bars should be sorted from highest to lowest and each bar should show its number.

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
library(ggplot2)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"))

Rows: 125 Columns: 14  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (7): case, month, location, summary, location\_type, race, prior\_mental\_i...  
dbl (6): year, day, fatalities, injured, total\_victims, age\_of\_shooter  
lgl (1): male  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Group data by race and count the number of cases  
shooters\_by\_race <- mass\_shootings %>%  
 group\_by(race) %>%  
 summarise(Number\_of\_Mass\_Shooters = n()) %>%  
 arrange(desc(Number\_of\_Mass\_Shooters)) # Arrange in descending order  
  
# Generate bar chart  
ggplot(shooters\_by\_race, aes(x = reorder(race, Number\_of\_Mass\_Shooters), y = Number\_of\_Mass\_Shooters, fill = race)) +  
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = Number\_of\_Mass\_Shooters), vjust = -0.5) +  
 theme\_minimal() +  
 labs(x = "Race", y = "Number of Mass Shooters", title = "Number of Mass Shooters by Race") +  
 coord\_flip() # Flip the chart to horizontal orientation

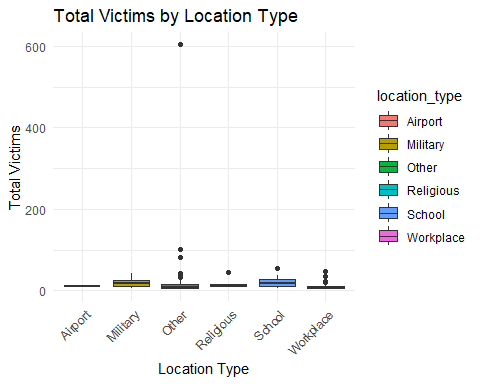


* Generate a boxplot visualizing the number of total victims, by type of location.

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
library(ggplot2)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"))

Rows: 125 Columns: 14  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (7): case, month, location, summary, location\_type, race, prior\_mental\_i...  
dbl (6): year, day, fatalities, injured, total\_victims, age\_of\_shooter  
lgl (1): male  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Generate boxplot  
ggplot(mass\_shootings, aes(x = location\_type, y = total\_victims, fill = location\_type)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(x = "Location Type", y = "Total Victims", title = "Total Victims by Location Type") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability

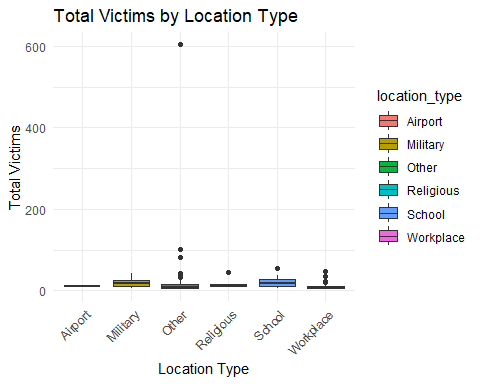


* Redraw the same plot, but remove the Las Vegas Strip massacre from the dataset.

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
library(ggplot2)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"))

Rows: 125 Columns: 14  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (7): case, month, location, summary, location\_type, race, prior\_mental\_i...  
dbl (6): year, day, fatalities, injured, total\_victims, age\_of\_shooter  
lgl (1): male  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Generate boxplot  
ggplot(mass\_shootings, aes(x = location\_type, y = total\_victims, fill = location\_type)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(x = "Location Type", y = "Total Victims", title = "Total Victims by Location Type") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability



### More open-ended questions

Address the following questions. Generate appropriate figures/tables to support your conclusions.

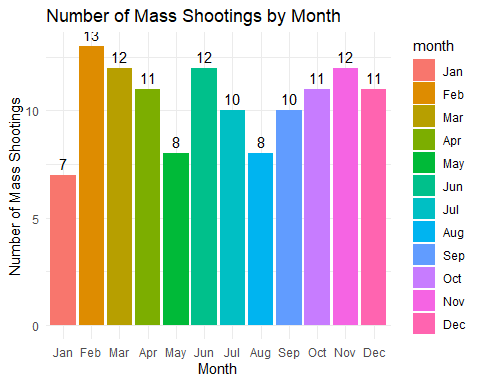
* How many white males with prior signs of mental illness initiated a mass shooting after 2000?

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"), show\_col\_types = FALSE)  
  
# Filter the data and count the number of incidents  
white\_males\_mental\_illness\_post\_2000 <- mass\_shootings %>%  
 filter(year > 2000, male == TRUE, race == "White", prior\_mental\_illness == "Yes") %>%  
 nrow() # Count the number of rows  
  
# Print the result  
print(paste("The number of white males with prior signs of mental illness who initiated a mass shooting after 2000 is:", white\_males\_mental\_illness\_post\_2000))

[1] "The number of white males with prior signs of mental illness who initiated a mass shooting after 2000 is: 22"

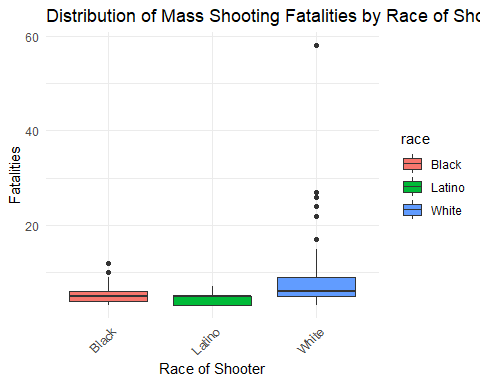
* Which month of the year has the most mass shootings? Generate a bar chart sorted in chronological (natural) order (Jan-Feb-Mar- etc) to provide evidence of your answer.

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
library(ggplot2)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"), show\_col\_types = FALSE)  
  
# Count the number of shootings in each month  
shootings\_by\_month <- mass\_shootings %>%  
 group\_by(month) %>%  
 summarise(Number\_of\_Mass\_Shootings = n())   
  
# Order the data by month  
months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")  
shootings\_by\_month$month <- factor(shootings\_by\_month$month, levels = months)  
  
# Generate bar chart  
ggplot(shootings\_by\_month, aes(x = month, y = Number\_of\_Mass\_Shootings, fill = month)) +  
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = Number\_of\_Mass\_Shootings), vjust = -0.5) +  
 theme\_minimal() +  
 labs(x = "Month", y = "Number of Mass Shootings", title = "Number of Mass Shootings by Month")



* How does the distribution of mass shooting fatalities differ between White and Black shooters? What about White and Latino shooters?

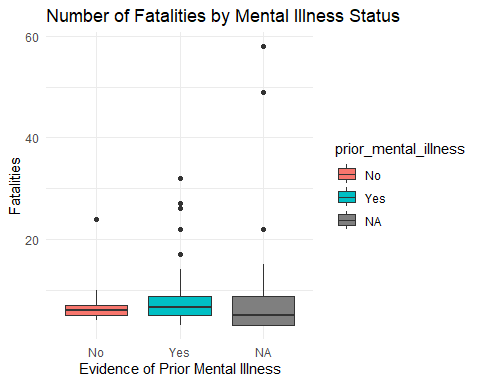
# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
library(ggplot2)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"), show\_col\_types = FALSE)  
  
# Filter the data to include only shootings with White, Black, or Latino shooters  
filtered\_data <- mass\_shootings %>% filter(race %in% c("White", "Black", "Latino"))  
  
# Generate boxplots  
ggplot(filtered\_data, aes(x = race, y = fatalities, fill = race)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(x = "Race of Shooter", y = "Fatalities", title = "Distribution of Mass Shooting Fatalities by Race of Shooter") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability



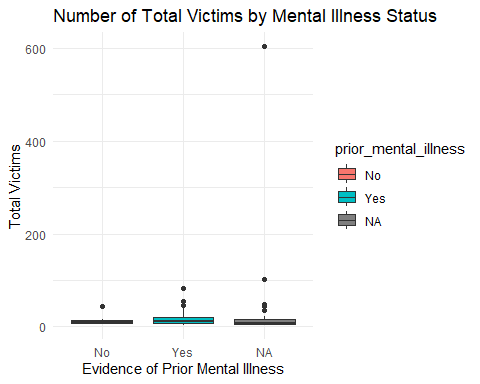
### Very open-ended

* Are mass shootings with shooters suffering from mental illness different from mass shootings with no signs of mental illness in the shooter?

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
library(ggplot2)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"), show\_col\_types = FALSE)  
  
# Boxplot for fatalities  
ggplot(mass\_shootings, aes(x = prior\_mental\_illness, y = fatalities, fill = prior\_mental\_illness)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(x = "Evidence of Prior Mental Illness", y = "Fatalities", title = "Number of Fatalities by Mental Illness Status")



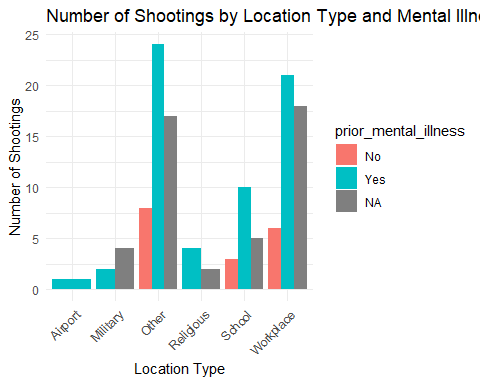
# Boxplot for total victims  
ggplot(mass\_shootings, aes(x = prior\_mental\_illness, y = total\_victims, fill = prior\_mental\_illness)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(x = "Evidence of Prior Mental Illness", y = "Total Victims", title = "Number of Total Victims by Mental Illness Status")



# Create a summarized data frame for location data  
location\_data <- mass\_shootings %>%  
 group\_by(location\_type, prior\_mental\_illness) %>%  
 summarise(Number\_of\_Shootings = n())

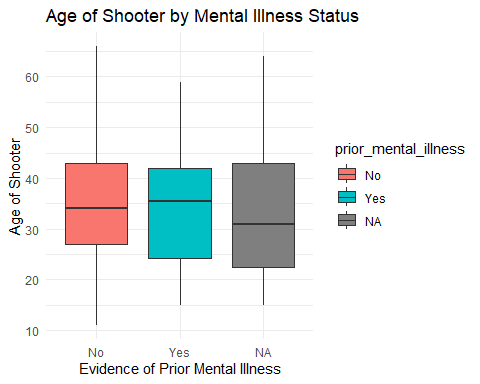
`summarise()` has grouped output by 'location\_type'. You can override using the  
`.groups` argument.

# Bar chart for location types  
ggplot(location\_data, aes(x = location\_type, y = Number\_of\_Shootings, fill = prior\_mental\_illness)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 theme\_minimal() +  
 labs(x = "Location Type", y = "Number of Shootings", title = "Number of Shootings by Location Type and Mental Illness Status") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability



# Boxplot for age of shooter  
ggplot(mass\_shootings, aes(x = prior\_mental\_illness, y = age\_of\_shooter, fill = prior\_mental\_illness)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(x = "Evidence of Prior Mental Illness", y = "Age of Shooter", title = "Age of Shooter by Mental Illness Status")

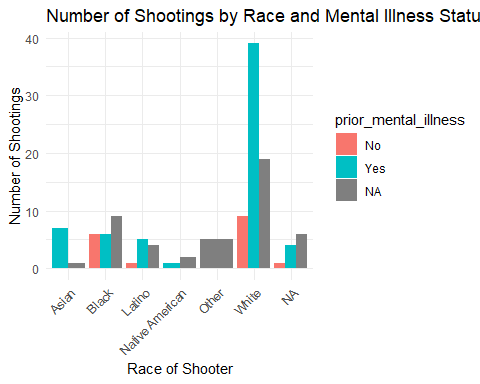
Warning: Removed 3 rows containing non-finite values (`stat\_boxplot()`).



# Create a summarized data frame for race data  
race\_data <- mass\_shootings %>%  
 group\_by(race, prior\_mental\_illness) %>%  
 summarise(Number\_of\_Shootings = n())

`summarise()` has grouped output by 'race'. You can override using the  
`.groups` argument.

# Bar chart for race of shooter  
ggplot(race\_data, aes(x = race, y = Number\_of\_Shootings, fill = prior\_mental\_illness)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 theme\_minimal() +  
 labs(x = "Race of Shooter", y = "Number of Shootings", title = "Number of Shootings by Race and Mental Illness Status") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability



There seems to be a higher incidence of mass shootings perpetrated by individuals who show evidence of prior mental illness compared to those who do not or for whom we have no data on their mental health status.

A noteworthy observation is that the proportion of white male shooters with a history of mental illness is significantly higher compared to other racial groups. This discrepancy is quite pronounced, with the exception of the Asian demographic, where the total number of incidents is relatively low (less than 10).

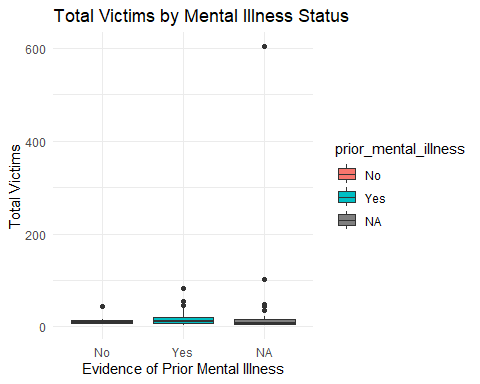
In terms of the age of the shooters, there appears to be no significant difference across categories based on the presence or absence of mental illness, or the lack of data on mental health status. This suggests that age may not be a significant factor in these incidents when considered in relation to mental health status.

When examining the type of locations where these incidents occur, it is observed that individuals with prior mental illness are involved in more mass shootings across almost all locations, with the exception of military locations. There could be several explanations for this exception:

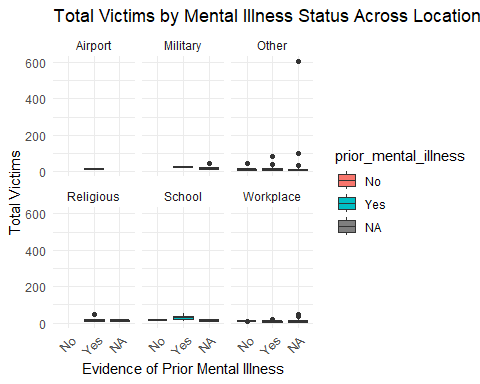
1. Lack of data: The mental health status of shooters in military locations might not be accurately recorded or disclosed, leading to an underreporting of incidents involving individuals with mental illness.
2. Access to weapons: In a military environment, there is typically easy access to firearms. This could potentially facilitate impulsive acts of violence, with less opportunity for individuals to reconsider or deescalate their actions.

* Assess the relationship between mental illness and total victims, mental illness and location type, and the intersection of all three variables.

# Loading necessary packages  
library(readr)  
library(dplyr)  
library(here)  
library(ggplot2)  
  
# Load the CSV data  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"), show\_col\_types = FALSE)  
  
# Boxplot for Mental Illness and Total Victims  
ggplot(mass\_shootings, aes(x = prior\_mental\_illness, y = total\_victims, fill = prior\_mental\_illness)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(x = "Evidence of Prior Mental Illness", y = "Total Victims", title = "Total Victims by Mental Illness Status")



# Intersection of Mental Illness, Total Victims, and Location Type  
ggplot(mass\_shootings, aes(x = prior\_mental\_illness, y = total\_victims, fill = prior\_mental\_illness)) +  
 geom\_boxplot() +  
 facet\_wrap(~location\_type) +  
 theme\_minimal() +  
 labs(x = "Evidence of Prior Mental Illness", y = "Total Victims", title = "Total Victims by Mental Illness Status Across Location Types") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability



The boxplots reveal that incidents involving shooters with prior mental illness, or cases where such data is not available (NA), tend to show a higher number of outliers in terms of total victims compared to incidents where the shooter has no reported signs of mental illness. This could potentially indicate a higher degree of variability in the behavior and actions of shooters with mental illness, which in turn could lead to a wider range of victim totals in the incidents they are involved in.

Furthermore, when considering the intersection of mental illness, total victims, and location type, we observe interesting patterns. In highly secured locations, such as airports and military bases where the presence of security personnel and firearms is substantial, the number of victims in mass shooting incidents tends to be lower and with less outliers. This could possibly be attributed to the quick intervention capabilities of security forces in such environments, thus limiting the extent of these tragedies.

# Exploring credit card fraud

We will be using a dataset with credit card transactions containing legitimate and fraud transactions. Fraud is typically well below 1% of all transactions, so a naive model that predicts that all transactions are legitimate and not fraudulent would have an accuracy of well over 99%– pretty good, no? (well, not quite as we will see later in the course)

You can read more on credit card fraud on [Credit Card Fraud Detection Using Weighted Support Vector Machine](https://www.scirp.org/journal/paperinformation.aspx?paperid=105944)

The dataset we will use consists of credit card transactions and it includes information about each transaction including customer details, the merchant and category of purchase, and whether or not the transaction was a fraud.

## Obtain the data

The dataset is too large to be hosted on Canvas or Github, so please download it from dropbox https://www.dropbox.com/sh/q1yk8mmnbbrzavl/AAAxzRtIhag9Nc\_hODafGV2ka?dl=0 and save it in your dsb repo, under the data folder

Rows: 671,028  
Columns: 14  
$ trans\_date\_trans\_time <dttm> 2019-02-22 07:32:58, 2019-02-16 15:07:20, 2019-…  
$ trans\_year <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2020, …  
$ category <chr> "entertainment", "kids\_pets", "personal\_care", "…  
$ amt <dbl> 7.79, 3.89, 8.43, 40.00, 54.04, 95.61, 64.95, 3.…  
$ city <chr> "Veedersburg", "Holloway", "Arnold", "Apison", "…  
$ state <chr> "IN", "OH", "MO", "TN", "CO", "GA", "MN", "AL", …  
$ lat <dbl> 40.1186, 40.0113, 38.4305, 35.0149, 39.4584, 32.…  
$ long <dbl> -87.2602, -80.9701, -90.3870, -85.0164, -106.385…  
$ city\_pop <dbl> 4049, 128, 35439, 3730, 277, 1841, 136, 190178, …  
$ job <chr> "Development worker, community", "Child psychoth…  
$ dob <date> 1959-10-19, 1946-04-03, 1985-03-31, 1991-01-28,…  
$ merch\_lat <dbl> 39.41679, 39.74585, 37.73078, 34.53277, 39.95244…  
$ merch\_long <dbl> -87.52619, -81.52477, -91.36875, -84.10676, -106…  
$ is\_fraud <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

The data dictionary is as follows

| column(variable) | description |
| --- | --- |
| trans\_date\_trans\_time | Transaction DateTime |
| trans\_year | Transaction year |
| category | category of merchant |
| amt | amount of transaction |
| city | City of card holder |
| state | State of card holder |
| lat | Latitude location of purchase |
| long | Longitude location of purchase |
| city\_pop | card holder’s city population |
| job | job of card holder |
| dob | date of birth of card holder |
| merch\_lat | Latitude Location of Merchant |
| merch\_long | Longitude Location of Merchant |
| is\_fraud | Whether Transaction is Fraud (1) or Not (0) |

* In this dataset, how likely are fraudulent transactions? Generate a table that summarizes the number and frequency of fraudulent transactions per year.

# Load the dplyr package  
library(dplyr)  
  
# Generate a table summarizing fraudulent transactions per year  
fraud\_per\_year <- card\_fraud %>%  
 # Group the data by year and whether or not the transaction was fraudulent  
 group\_by(trans\_year, is\_fraud) %>%  
 # Create summary statistics for each group: count of transactions and percentage of total transactions  
 summarise(  
 count = n(),   
 percentage = count / nrow(card\_fraud) \* 100,  
 .groups = "drop" # This returns to ungrouped data after summarising  
 ) %>%  
 # Filter the data to show only fraudulent transactions (where is\_fraud == 1)  
 filter(is\_fraud == 1)  
  
# Print the table  
print(fraud\_per\_year)

# A tibble: 2 × 4  
 trans\_year is\_fraud count percentage  
 <dbl> <dbl> <int> <dbl>  
1 2019 1 2721 0.405  
2 2020 1 1215 0.181

* How much money (in US$ terms) are fraudulent transactions costing the company? Generate a table that summarizes the total amount of legitimate and fraudulent transactions per year and calculate the % of fraudulent transactions, in US$ terms.

# Load the dplyr package  
library(dplyr)  
  
# Generate a table summarizing the total transaction amount by fraud status per year  
total\_amt\_per\_year <- card\_fraud %>%  
 # Group the data by year and whether or not the transaction was fraudulent  
 group\_by(trans\_year, is\_fraud) %>%  
 # Summarise the total transaction amount for each group  
 summarise(  
 total\_amt = sum(amt, na.rm = TRUE),   
 .groups = "drop" # This returns to ungrouped data after summarising  
 )  
  
# Calculate the percentage of fraudulent transactions in terms of the total transaction amount  
total\_amt\_per\_year <- total\_amt\_per\_year %>%  
 # Compute the total transaction amount per year (including both fraudulent and legitimate transactions)  
 mutate(total\_amt\_year = sum(total\_amt)) %>%  
 # Calculate the percentage of the total transaction amount that is fraudulent  
 mutate(percentage\_fraud = total\_amt / total\_amt\_year \* 100 \* is\_fraud) %>%  
 # Filter the data to show only fraudulent transactions (where is\_fraud == 1)  
 filter(is\_fraud == 1)  
  
# Print the table  
print(total\_amt\_per\_year)

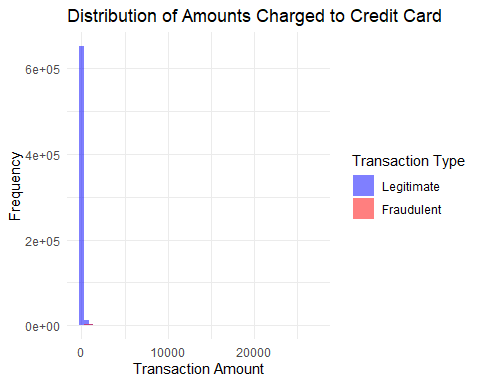
# A tibble: 2 × 5  
 trans\_year is\_fraud total\_amt total\_amt\_year percentage\_fraud  
 <dbl> <dbl> <dbl> <dbl> <dbl>  
1 2019 1 1423140. 47183904. 3.02  
2 2020 1 651949. 47183904. 1.38

* Generate a histogram that shows the distribution of amounts charged to credit card, both for legitimate and fraudulent accounts. Also, for both types of transactions, calculate some quick summary statistics.

# Load the necessary libraries  
library(dplyr)  
library(ggplot2)  
  
# Generate summary statistics for legitimate and fraudulent transactions  
summary\_stats <- card\_fraud %>%  
 group\_by(is\_fraud) %>%  
 summarise(  
 mean = mean(amt, na.rm = TRUE),  
 median = median(amt, na.rm = TRUE),  
 min = min(amt, na.rm = TRUE),  
 max = max(amt, na.rm = TRUE),  
 .groups = "drop" # This returns to ungrouped data after summarising  
 )  
  
# Print the summary statistics  
print(summary\_stats)

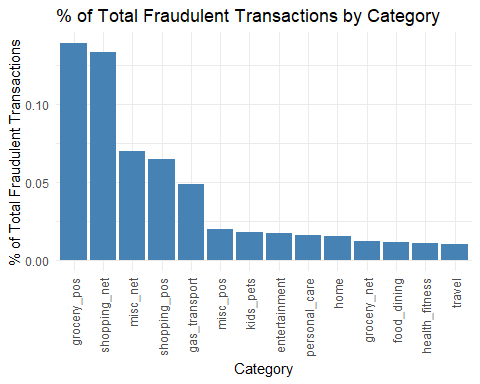
# A tibble: 2 × 5  
 is\_fraud mean median min max  
 <dbl> <dbl> <dbl> <dbl> <dbl>  
1 0 67.6 47.2 1 27120.  
2 1 527. 369. 1.06 1334.

# Generate histograms for legitimate and fraudulent transactions  
ggplot(card\_fraud, aes(x = amt, fill = as.factor(is\_fraud))) +  
 geom\_histogram(position = "identity", alpha = 0.5, bins = 50) +  
 scale\_fill\_manual(values = c("blue", "red"),  
 labels = c("Legitimate", "Fraudulent"),  
 name = "Transaction Type") +  
 theme\_minimal() +  
 labs(title = "Distribution of Amounts Charged to Credit Card",  
 x = "Transaction Amount",  
 y = "Frequency")



* What types of purchases are most likely to be instances of fraud? Consider category of merchants and produce a bar chart that shows % of total fraudulent transactions sorted in order.

# Load the necessary libraries  
library(dplyr)  
library(ggplot2)  
  
# Calculate the number and percentage of fraudulent transactions per category  
fraud\_by\_category <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 group\_by(category) %>%  
 summarise(  
 count = n(),  
 percentage = count / nrow(card\_fraud) \* 100,  
 .groups = "drop" # This returns to ungrouped data after summarising  
 ) %>%  
 arrange(desc(percentage)) # Sort by percentage in descending order  
  
# Generate a bar chart of the percentage of fraudulent transactions per category  
ggplot(fraud\_by\_category, aes(x = reorder(category, -percentage), y = percentage)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1)) +  
 labs(  
 title = "% of Total Fraudulent Transactions by Category",  
 x = "Category",  
 y = "% of Total Fraudulent Transactions"  
 )



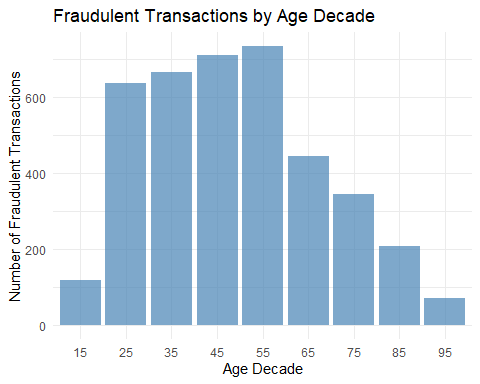
* When is fraud more prevalent? Which days, months, hours? To create new variables to help you in your analysis, we use the lubridate package and the following code

mutate(  
 date\_only = lubridate::date(trans\_date\_trans\_time),  
 month\_name = lubridate::month(trans\_date\_trans\_time, label=TRUE),  
 hour = lubridate::hour(trans\_date\_trans\_time),  
 weekday = lubridate::wday(trans\_date\_trans\_time, label = TRUE)  
 )

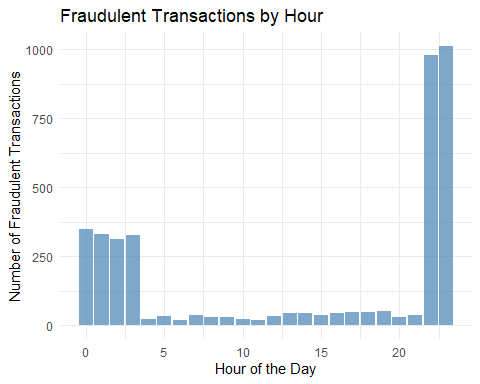
* Are older customers significantly more likely to be victims of credit card fraud? To calculate a customer’s age, we use the lubridate package and the following code

mutate(  
 age = interval(dob, trans\_date\_trans\_time) / years(1),  
 )

# Load necessary libraries  
library(dplyr)  
library(ggplot2)  
library(lubridate)  
  
# Create new variables for date\_only, month\_name, hour, weekday, and age  
card\_fraud <- card\_fraud %>%  
 mutate(  
 date\_only = lubridate::date(trans\_date\_trans\_time),  
 month\_name = lubridate::month(trans\_date\_trans\_time, label = TRUE),  
 hour = lubridate::hour(trans\_date\_trans\_time),  
 weekday = lubridate::wday(trans\_date\_trans\_time, label = TRUE),  
 dob = as.Date(dob), # ensure dob is in date format  
 age = as.numeric(date\_only - dob) / 365.25, # calculate age in years  
 age\_decade = cut(age, breaks = seq(0, 100, 10), labels = seq(10, 100, 10) - 5, include.lowest = TRUE) # bin age into decades  
 )  
  
# Analyze fraudulent transactions by age\_decade  
fraud\_by\_age\_decade <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 group\_by(age\_decade) %>%  
 summarise(  
 count = n(),  
 .groups = "drop" # This returns to ungrouped data after summarising  
 )  
  
# Generate a bar chart for fraudulent transactions by age\_decade  
ggplot(fraud\_by\_age\_decade, aes(x = age\_decade, y = count)) +  
 geom\_bar(stat = "identity", fill = "steelblue", alpha = 0.7) +  
 theme\_minimal() +  
 labs(  
 title = "Fraudulent Transactions by Age Decade",  
 x = "Age Decade",  
 y = "Number of Fraudulent Transactions"  
 )

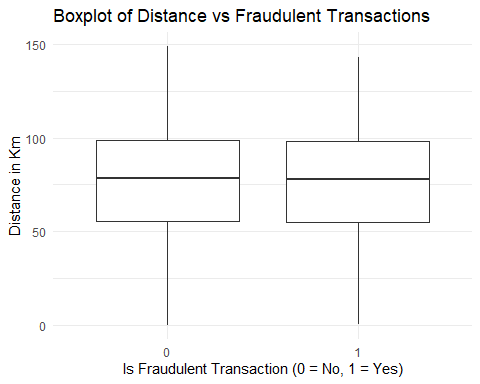


# Analyze fraudulent transactions by hour  
fraud\_by\_hour <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 group\_by(hour) %>%  
 summarise(  
 count = n(),  
 .groups = "drop" # This returns to ungrouped data after summarising  
 )  
  
# Generate a bar chart for fraudulent transactions by hour  
ggplot(fraud\_by\_hour, aes(x = hour, y = count)) +  
 geom\_bar(stat = "identity", fill = "steelblue", alpha = 0.7) +  
 theme\_minimal() +  
 labs(  
 title = "Fraudulent Transactions by Hour",  
 x = "Hour of the Day",  
 y = "Number of Fraudulent Transactions"  
 )



* Is fraud related to distance? The distance between a card holder’s home and the location of the transaction can be a feature that is related to fraud. To calculate distance, we need the latidue/longitude of card holders’s home and the latitude/longitude of the transaction, and we will use the [Haversine formula](https://en.wikipedia.org/wiki/Haversine_formula) to calculate distance. I adapted code to [calculate distance between two points on earth](https://www.geeksforgeeks.org/program-distance-two-points-earth/amp/) which you can find below

# Load required packages  
library(dplyr)  
library(ggplot2)  
  
# distance between card holder's home and transaction  
# code adapted from https://www.geeksforgeeks.org/program-distance-two-points-earth/amp/  
  
card\_fraud <- card\_fraud %>%  
 mutate(  
   
 # convert latitude/longitude to radians  
 lat1\_radians = lat / 57.29577951,  
 lat2\_radians = merch\_lat / 57.29577951,  
 long1\_radians = long / 57.29577951,  
 long2\_radians = merch\_long / 57.29577951,  
   
 # calculate distance in miles  
 distance\_miles = 3963.0 \* acos((sin(lat1\_radians) \* sin(lat2\_radians)) + cos(lat1\_radians) \* cos(lat2\_radians) \* cos(long2\_radians - long1\_radians)),  
  
 # calculate distance in km  
 distance\_km = 6377.830272 \* acos((sin(lat1\_radians) \* sin(lat2\_radians)) + cos(lat1\_radians) \* cos(lat2\_radians) \* cos(long2\_radians - long1\_radians))  
   
 )   
  
# Create a boxplot of distance\_km in relation to is\_fraud  
  
ggplot(card\_fraud, aes(x = factor(is\_fraud), y = distance\_km)) +  
 geom\_boxplot() +  
 labs(title = "Boxplot of Distance vs Fraudulent Transactions",  
 x = "Is Fraudulent Transaction (0 = No, 1 = Yes)",  
 y = "Distance in Km") +  
 theme\_minimal()



Plot a boxplot or a violin plot that looks at the relationship of distance and is\_fraud. Does distance seem to be a useful feature in explaining fraud?

No it is not. No signifcant difference in distance between a fraud to non-fraud.

# Exploring sources of electricity production, CO2 emissions, and GDP per capita.

There are many sources of data on how countries generate their electricity and their CO2 emissions. I would like you to create three graphs:

## 1. A stacked area chart that shows how your own country generated its electricity since 2000.

You will use

geom\_area(colour="grey90", alpha = 0.5, position = "fill")

## 2. A scatter plot that looks at how CO2 per capita and GDP per capita are related

## 3. A scatter plot that looks at how electricity usage (kWh) per capita/day GDP per capita are related

We will get energy data from the Our World in Data website, and CO2 and GDP per capita emissions from the World Bank, using the wbstatspackage.

library(tidyverse)  
library(wbstats)  
library(countrycode)  
library(patchwork)  
  
# Download electricity data  
url <- "https://nyc3.digitaloceanspaces.com/owid-public/data/energy/owid-energy-data.csv"  
  
energy <- read\_csv(url) %>%   
 filter(year >= 1990) %>%   
 drop\_na(iso\_code) %>%   
 select(1:3,  
 biofuel = biofuel\_electricity,  
 coal = coal\_electricity,  
 gas = gas\_electricity,  
 hydro = hydro\_electricity,  
 nuclear = nuclear\_electricity,  
 oil = oil\_electricity,  
 other\_renewable = other\_renewable\_exc\_biofuel\_electricity,  
 solar = solar\_electricity,  
 wind = wind\_electricity,   
 electricity\_demand,  
 electricity\_generation,  
 net\_elec\_imports,  
 energy\_per\_capita,  
 energy\_per\_gdp,  
 per\_capita\_electricity  
 )   
  
# Turn energy to long, tidy format  
energy <- energy %>%  
 pivot\_longer(  
 cols = c(biofuel, coal, gas, hydro, nuclear, oil, other\_renewable, solar, wind),  
 names\_to = "source",  
 values\_to = "electricity\_generated"  
 )  
  
# Download data for C02 emissions per capita https://data.worldbank.org/indicator/EN.ATM.CO2E.PC  
co2\_percap <- wb\_data(country = "countries\_only",   
 indicator = "EN.ATM.CO2E.PC",   
 start\_date = 1990,   
 end\_date = 2022,  
 return\_wide=FALSE) %>%   
 filter(!is.na(value)) %>%   
 select(-c(unit, obs\_status, footnote, last\_updated)) %>%   
 rename(year = date,  
 co2percap = value,  
 iso\_code = iso3c) %>%  
 mutate(year = as.numeric(year))  
  
# Download data for GDP per capita https://data.worldbank.org/indicator/NY.GDP.PCAP.PP.KD  
gdp\_percap <- wb\_data(country = "countries\_only",   
 indicator = "NY.GDP.PCAP.PP.KD",   
 start\_date = 1990,   
 end\_date = 2022,  
 return\_wide=FALSE) %>%   
 filter(!is.na(value)) %>%   
 select(-c(unit, obs\_status, footnote, last\_updated)) %>%   
 rename(year = date,  
 GDPpercap = value,  
 iso\_code = iso3c) %>%  
 mutate(year = as.numeric(year))  
  
# Merge the dataframes  
merged\_data <- energy %>%  
 left\_join(co2\_percap, by = c("iso\_code", "year")) %>%  
 left\_join(gdp\_percap, by = c("iso\_code", "year"))  
  
# Function to create plots  
plot\_country\_data <- function(country\_name) {  
 # Filter the data for the country  
 country\_data <- merged\_data %>%  
 filter(country == country\_name)  
   
 # Define your three plots here  
 plot1 <- ggplot(country\_data, aes(x = year, y = electricity\_generated, fill = source)) +  
 geom\_area(colour="grey90", alpha = 0.5, position = "fill") +  
 labs(x = "Year", y = "Electricity Generated", fill = "Source", title = paste("Electricity Generation in", country\_name))  
  
 plot2 <-  
 ggplot(country\_data, aes(x = GDPpercap, y = co2percap)) +  
 geom\_point() +  
 labs(x = "GDP per Capita", y = "CO2 Emissions per Capita", title = paste("CO2 Emissions vs GDP in", country\_name))  
  
 plot3 <- ggplot(country\_data, aes(x = GDPpercap, y = per\_capita\_electricity)) +  
 geom\_point() +  
 labs(x = "GDP per Capita", y = "Electricity Usage per Capita", title = paste("Electricity Usage vs GDP in", country\_name))  
  
 # Combine the plots  
 combined\_plot <- plot1 / plot2 / plot3  
  
 return(combined\_plot)  
}  
  
# You can now call the function with the country name to get the plots. For example:  
# plot\_country\_data("USA")

Specific questions: ansers on the code

1. How would you turn energy to long, tidy format?

the **pivot\_longer()** function is used to transform the **energy** data from wide to long format. This is done for the energy sources columns (biofuel, coal, gas, hydro, nuclear, oil, other\_renewable, solar, wind). This step is crucial because it makes the data tidy, which is easier to manipulate, model, and visualize.

Here’s the specific code:

energy <- energy %>% pivot\_longer( cols = c(biofuel, coal, gas, hydro, nuclear, oil, other\_renewable, solar, wind), names\_to = “source”, values\_to = “electricity\_generated” )

1. You may need to join these data frames
   * Use left\_join from dplyr to [join the tables](http://r4ds.had.co.nz/relational-data.html)
   * To complete the merge, you need a unique *key* to match observations between the data frames. Country names may not be consistent among the three dataframes, so please use the 3-digit ISO code for each country
   * An aside: There is a great package called [countrycode](https://github.com/vincentarelbundock/countrycode) that helps solve the problem of inconsistent country names (Is it UK? United Kingdon? Great Britain?). countrycode() takes as an input a country’s name in a specific format and outputs it using whatever format you specify.

To combine the **energy**, **co2\_percap**, and **gdp\_percap** data frames, the **left\_join()** function from the **dplyr** package is used. This function performs a left join, which keeps all rows from the left data frame (**energy** in this case) and appends matching rows from the right data frame.

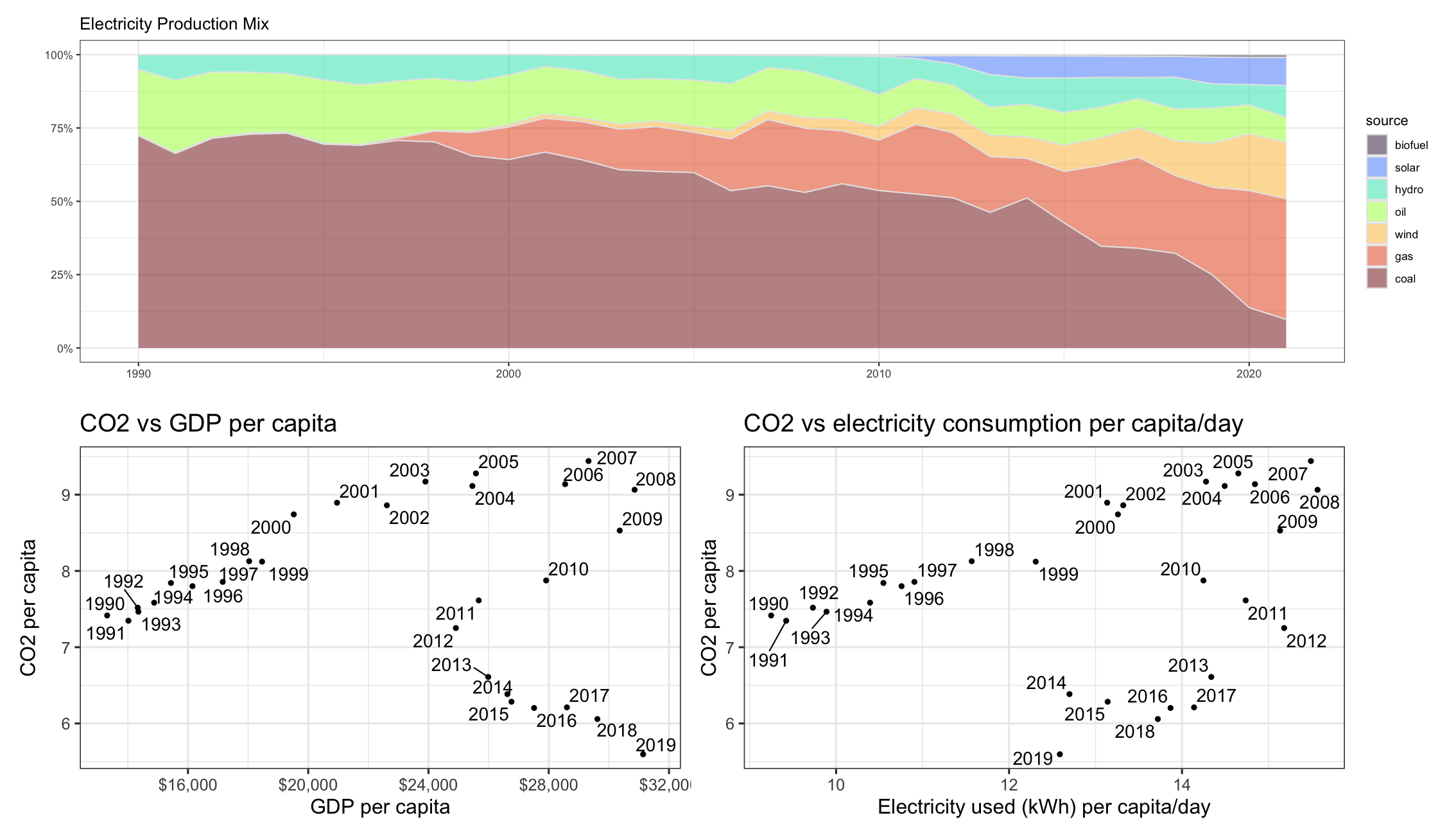
In order to match rows between the data frames, the **iso\_code** (3-digit ISO code for each country) and **year** are used as keys, because country names can sometimes be inconsistent across different datasets.

merged\_data <- energy %>%

left\_join(co2\_percap, by = c(“iso\_code”, “year”)) %>%

left\_join(gdp\_percap, by = c(“iso\_code”, “year”))

1. Write a function that takes as input any country’s name and returns all three graphs. You can use the patchwork package to arrange the three graphs as shown below



plot\_country\_data <- function(country\_name) {

# Filter the data for the country

country\_data <- merged\_data %>%

filter(country == country\_name)

# Define your three plots here

plot1 <- ggplot(country\_data, aes(x = year, y = electricity\_generated, fill = source)) +

geom\_area(colour=“grey90”, alpha = 0.5, position = “fill”) +

labs(x = “Year”, y = “Electricity Generated”, fill = “Source”, title = paste(“Electricity Generation in”, country\_name))

plot2 <- ggplot(country\_data, aes(x = GDPpercap, y = co2percap)) +

geom\_point() +

labs(x = “GDP per Capita”, y = “CO2 Emissions per Capita”, title = paste(“CO2 Emissions vs GDP in”, country\_name))

plot3 <- ggplot(country\_data, aes(x = GDPpercap, y = per\_capita\_electricity)) +

geom\_point() +

labs(x = “GDP per Capita”, y = “Electricity Usage per Capita”, title = paste(“Electricity Usage vs GDP in”, country\_name))

# Combine the plots

combined\_plot <- plot1 / plot2 / plot3

return(combined\_plot)

}

# Deliverables

There is a lot of explanatory text, comments, etc. You do not need these, so delete them and produce a stand-alone document that you could share with someone. Knit the edited and completed Quarto Markdown (qmd) file as a Word document (use the “Render” button at the top of the script editor window) and upload it to Canvas. You must be commiting and pushing tour changes to your own Github repo as you go along.

# Details

* Who did you collaborate with: Sean O’Doherty
* Approximately how much time did you spend on this problem set: 3h
* What, if anything, gave you the most trouble: The last part

As a true test to yourself, do you understand the code you submitted and are you able to explain it to someone else?